## CLAIMS

**We claim:**

1. A method for accessing status information related to a process that is executable by one or more nodes from over a network, the method comprising:

retrieving status information related to an executable process by a process management system executing on a primary node; and

storing status information related to the executable process into a data structure, wherein the data structure is available to any node capable of accessing the process management system.

2. A computer-readable medium having stored thereon computer-executable instructions for performing the method of claim 1.

3. The method of claim 1, further comprising: invoking one or more script engines by the process management system to execute script code that performs the action of the executable process, wherein the process management system handles multiple script threads during the execution of the process.

4. The method of claim 3, wherein the one or more script engines are maintained by a process management system that executes on the one or more nodes.

5. The method of claim 4, wherein the one or more nodes include the primary node.

6. The method of claim 1, wherein the step of retrieving includes polling the process management system on the one or more nodes to obtain status information related to the executable process.

7. The method of claim 6, wherein the step of polling is performed by the process management system residing on the primary node over an established connection with the one or more nodes.

8. The method of claim 7, wherein the one or more nodes include the primary node.

9. The method of claim 1, wherein the step of retrieving includes receiving status information from the one or more nodes that are polled by the process management system.

10. The method of claim 9, wherein the status information is generated by script code that is executed by one or more script engines that reside on the one or more nodes.

11. The method of claim 10, wherein the one or more nodes include the primary node.

12. The method of claim 1, wherein the step of storing is performed by the process management system executing on the primary node.

13. The method of claim 1, wherein the step of storing further includes:

placing the status information relative to the executable process into a private data structure by the process management system on the primary node, wherein the data structure is accessible to only the script threads that are spawned during the execution of the process.

14. The method of claim 1, wherein the step of storing further includes:

placing the status information relative to the executable process into a status value data structure that is accessible to any node capable of accessing the process management system executing on the primary node.

15. The system of claim 16, wherein the status value data structure comprises data for providing an indication of an event that occurs during the execution of the process.

16. The method of claim 1, further comprising: establishing a connection between a process management system executing on the one or more nodes and the process management system residing on the primary node, wherein the connection is established by the script code in execution by the one or more script engines.

17. The method of claim 1, further comprising:

establishing a connection between one or more nodes and the process management system residing on the primary node, wherein the connection is established from a user interface executing on the one or more nodes; and

accessing the process management system from over the established connection by the user interface executing on the one or more nodes.

18. The method of claim 17, wherein the step of establishing includes accepting a command as input by the user interface to establish a connection with the process management system executing on the primary node.

19. The method of claim 17, wherein the step of accessing includes accepting a command as input by the user interface to invoke the action of the executable process by the process management system from over the established connection.

20. The method of claim 17, wherein the step of accessing includes accepting a command as input by the user interface to poll the process management system for status information from over the established connection.

21. The method of claim 17, wherein the user interface receives messages from the process management system over the established connection.

22. The method of claim 21, wherein the messages contain information that is descriptive of the primary node.

23. The method of claim 21, wherein the messages contain information that is descriptive of a particular event that occurs during the execution of the process.

24. The method of claim 21, wherein the messages contain a data structure that is generated as a result of the execution of the script code by the one or more script engines to indicate the status of the executable process.

25. A computer-readable medium having computer executable components comprising:

a first component for hosting one or more script engines, and for managing script threads that are spawned during the execution of a process;

a second component for handling the execution and monitoring of local processes that are launched during the execution of script code by the one or more script engines;

a third component for:

(i) receiving and accepting requests from one or more nodes to establish a connection over a network;

(ii) receiving commands from the one or more nodes from over an established connection to invoke the action of the executable process;

(iii) receiving commands from the one or more nodes from over the established connection to request for information that is descriptive of the primary node; and

(iv) sending messages to the one or more nodes from over the established connection in response to the requests and commands received from the one or more nodes.

a fourth component for:

(i) enabling messages to be passed between the first component and the third component;

(ii) launching the execution of a primary script file on behalf of the first component, wherein the primary script file spawns the execution of the one or more script engines in response to a request received over the established connection to invoke an executable process;

(iii) storing information related to one or more script threads and local processes in execution; and

(iv) storing status information relative to the executable process into a data structure that is accessible to the script threads.

26. The computer-readable medium of claim 25, wherein the first component further comprises: a primary script file for spawning the execution of any of the one or more script engines in response to a request to invoke the executable process, the request indicating a specific script file to execute..

27. The computer-readable medium of claim 26, wherein the primary script file is automatically loaded onto the primary node for execution when the process management system is initiated.

28. The computer-readable medium of claim 26, wherein a failure to load the primary script file onto the primary node results in the first component indicating failure and terminating the established connection.

29. The computer-readable medium of claim 26, wherein the request is made by one or more nodes over an established connection.

30. The computer-readable medium of claim 26, wherein the request is made by a local process over an established connection.

31. The computer-readable medium of claim 25, wherein the one or more script engines hosted by the first component terminate the executable process when an execution error occurs.

32. The computer-readable medium of claim 25, wherein the first component further comprises: a global object for extending the behavior of the one or more script engines,

the script engine obtaining an identifier from the global object to perform a specific task upon encountering the identifier during the execution of the script code and determining that it cannot be interpreted.

33. The computer-readable medium of claim 325, wherein the identifier is an executable script variable having instructions for performing a specific task.

34. The computer-readable medium of claim 32, wherein the global object implements a communication channel between the first component and the one or more script engines to enable the one or more script engines to interact with the first component.

35. The computer-readable medium of claim 32, wherein the global object implements a communication channel between the first component and the one or more script engines to enable the first component to interact with the one or more script engines.

36. The computer-readable medium of claim 25, wherein the second component launches local processes that are called by a script in execution using the global object, and reports the status information and data generated by the local processes back to the calling script.

37. The computer-readable medium of claim 25, wherein the fourth component receives information from the second component that is descriptive of the local processes in execution by the one or more script engines.

38. The computer-readable medium of claim 25, wherein the fourth component receives status information from the one or more script threads and stores the information into a public data structure that is accessible to the one or more nodes capable of establishing a connection with the first component.

39. The computer-readable medium of claim 25, wherein the fourth component receives status information from the one or more script threads and stores the information into a private data structure that is accessible to only the script threads in operation during process execution.

40. The computer-readable medium of claim 25, wherein the fourth component receives status information from the one or more script threads and stores the information into a status value data structure that is accessible to the one or more nodes capable of establishing a connection with the first component.

41. The computer-readable medium of claim 40, wherein the status value data structure contains data for providing an indication of an event that occurs during the execution of the process.

42. A system for accessing status information that is stored on a primary node, wherein the status information is related to a process that is executable by one or more nodes from over a network, the system comprising:

one or more user interfaces for invoking the executable process and retrieving status information generated by one or more script engines in execution from over a network;

a multiple threaded process management system executing on a primary node for collecting and storing status information related to the executable process from the one or more nodes;

at least one script engine maintained by the process management system for accessing and executing script code; and

at least one database having stored therein script code for enabling the executable process, wherein the database is accessible by the at least one script engine.

43. The system of claim 42, wherein the one or more user interfaces establish a connection over the network with the process management system executing on the primary node.

44. The system of claim 42, wherein the one or more user interfaces are executed by one or more nodes tied to the network.

45. The system of claim 44, wherein the one or more nodes include the primary node.

46. The system of claim 42, wherein the one or more user interfaces accept as input commands to establish a connection with the process management system executing on the primary node.

47. The method of claim 42, wherein the one or more user interfaces accept as input commands to invoke the action of the executable process by the process management system, and sends requests to invoke the action of the executable process to the process management system from over the established connection.

48. The method of claim 42, wherein the one or more user interfaces accept as input commands to poll the process management system for status information, and sends requests to poll the process management system for status information from over the established connection.

49. The system of claim 42, wherein the one or more user interfaces receive messages from the process management system over the established connection in response to the polling.

50. The method of claim 49, wherein the messages contain information that is descriptive of the primary node.

51. The method of claim 49, wherein the messages contain information that is descriptive of a particular event that occurs during the execution of the process.

52. The method of claim 49, wherein the messages contain a data structure that is generated as a result of the execution of the script code by the one or more script engines to indicate the status of the executable process.

53. The system of claim 42, wherein the process management system accepts connection requests from one or more user interfaces operating on one or more nodes from over the established connection.

54. The system of claim 53, wherein the one or more nodes include the primary node.

55. The system of claim 42, wherein the process management system receives requests to invoke the action of the executable process from the one or more nodes connected to the process management system.

56. The system of claim 42, wherein the process management system continuously polls the one or more nodes connected to the process management system to obtain status information related to the executable process.

57. The system of claim 42, wherein the process management system stores the information into a public data structure that is accessible to the one or more nodes capable of establishing a connection with the first component.

58. The system of claim 42, wherein the process management system stores the status information relative to the process into a private data structure that is accessible to only the script threads in operation during process execution.

59. The system of claim 42, wherein the process management system stores the status information relative to the executable process into a status value data structure that is accessible to the one or more nodes having access to the status information.

60. The system of claim 59, wherein the status value data structure contains data for providing an indication of a particular event that occurs during the execution of the process.

61. The system of claim 42, wherein the process management system receives requests for status information relative to the executable process from the one or more nodes connected to the process management system.

62. The system of claim 42, wherein the process management system sends the public data structure to the one or more nodes in response to the request.

63. The system of claim 42, wherein the process management system sends the status value data structure to the one or more nodes in response to the request.